

# Les sous-requêtes

# Sous-requêtes – Principe de base

## Résultat d'une requête SQL :

Le résultat d'une requête est toujours une table, éventuellement à une seule ligne et une seule colonne.

Exemple : Etudiant (NumEtu, Nom, Prenom)

SELECT \* FROM etudiant

(requête de type “table”)

NumEtu	Nom	Prenom
1234	Toto	Tati
etc.	etc.	etc.

SELECT COUNT(\*) as NB FROM etudiant

(requête de type “valeur”)

NB
18

# Sous-requêtes – Principe de base

**SQL** va permettre d'écrire des requêtes faisant **référence à des résultats (table ou valeur)** d'autres requêtes.

Ces requêtes seront alors **des sous-requêtes** de la requête principale, elles seront incluses dans la requête principale entre parenthèses.

# Sous-requêtes – Principe de base

## Utilisation d'une sous-requête de type valeur

Une sous-requête de type valeur peut être utilisée **partout où SQL permet d'utiliser une valeur** (i.e. une constante) :

- dans le **SELECT** (liste des éléments à afficher),
- dans les clauses **WHERE** ou **HAVING** (dans une condition).

## Utilisation d'une sous-requête de type table

Une sous-requête de type table peut être utilisée **partout où SQL permet d'utiliser une table** :

- dans le **FROM** (éventuellement avec jointures),
- dans les clauses **WHERE** ou **HAVING** (dans une condition avec l'opérateur **IN**).

## Exemple 1 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe.

## Exemple 1 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe.

Ce serait facile, si on disposait de... ?

## Exemple 1 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe.

Ce serait facile, si on disposait de... la moyenne générale du groupe !

# Exemple 1 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe.

Ce serait facile, si on disposait de... la moyenne générale du groupe !

En effet, si cette moyenne était, disons, moyGroupe, on écrirait simplement :

```
SELECT Nom, Prénom, Moyenne  
FROM Etudiants  
WHERE Moyenne >= moyGroupe
```



## Exemple 1 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe.

Ce serait facile, si on disposait de... la moyenne générale du groupe !

Eh bien, calculons cette moyenne...

## Exemple 1 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe.

Ce serait facile, si on disposait de... la moyenne générale du groupe !

Eh bien, calculons cette moyenne...

```
SELECT AVG(Moyenne) FROM Etudiants
```

## Exemple 1 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe.

Ce serait facile, si on disposait de... la moyenne générale du groupe !

Eh bien, calculons cette moyenne...

**SELECT AVG(Moyenne) FROM Etudiants**

Il n'y a plus qu'à remplacer, dans la requête précédente, **moyGroupe** par cette (sous-)requête :

```
SELECT Nom, Prenom, Moyenne
FROM Etudiants
WHERE Moyenne >= (SELECT AVG(Moyenne) FROM Etudiants)
```

## Exemple 2 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe, *en affichant aussi cette moyenne générale*.

## Exemple 2 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe, *en affichant aussi cette moyenne générale*.

Nous avons déjà les ingrédients nécessaires ! Il suffit en effet d'utiliser deux fois notre sous-requête précédente :

```
SELECT Nom, Prenom, Moyenne,  
       (SELECT AVG(Moyenne) FROM Etudiants) AS moyGroupe  
FROM Etudiants  
WHERE Moyenne >= (SELECT AVG(Moyenne) FROM Etudiants)
```

## Exemple 2 : sous-requêtes de type valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

On veut afficher la liste des étudiants dont la moyenne est supérieure ou égale à la moyenne générale du groupe, *en affichant aussi cette moyenne générale*.

Nous avons déjà les ingrédients nécessaires ! Il suffit en effet d'utiliser deux fois notre sous-requête précédente :

```
SELECT Nom, Prenom, Moyenne,  
       (SELECT AVG(Moyenne) FROM Etudiants) AS moyGroupe  
FROM Etudiants  
WHERE Moyenne >= (SELECT AVG(Moyenne) FROM Etudiants)
```

Remarquons ici que SQL est « assez malin » pour ne pas recalculer plusieurs fois la sous-requête...

## Exemple 3 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville.

## Exemple 3 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville.

Ce serait facile, si on disposait... ?



## Exemple 3 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville.

Ce serait facile, si on disposait... **du nombre d'étudiants dans chaque ville !**

## Exemple 3 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville.

Ce serait facile, si on disposait... **du nombre d'étudiants dans chaque ville !**

En effet, si cette table s'appelait, disons, T, on écrirait simplement (par exemple) :

```
SELECT MAX(T.NbEtds) FROM T
```

## Exemple 3 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville.

Ce serait facile, si on disposait... **du nombre d'étudiants dans chaque ville !**

Eh bien, calculons cette table (connaître les villes est inutile) :

```
SELECT COUNT(*) AS NbEtds FROM Etudiants GROUP BY NumVille
```

## Exemple 3 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville.

Ce serait facile, si on disposait... du nombre d'étudiants dans chaque ville !

Eh bien, calculons cette table (connaître les villes est inutile) :

```
SELECT COUNT(*) AS NbEtds FROM Etudiants GROUP BY NumVille
```

Il n'y a plus qu'à « remplacer » :

```
SELECT MAX(T.NbEtds)  
FROM (SELECT COUNT(*) AS NbEtds FROM Etudiants  
GROUP BY NumVille) AS T
```

## Exemple 3 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville.

Ce serait facile, si on disposait... du nombre d'étudiants dans chaque ville !

Eh bien, calculons cette table (connaître les villes est inutile) :

```
SELECT COUNT(*) AS NbEtds FROM Etudiants GROUP BY NumVille
```

Il n'y a plus qu'à « remplacer » :

```
SELECT MAX(T.NbEtds)  
FROM (SELECT COUNT(*) AS NbEtds FROM Etudiants  
      GROUP BY NumVille) AS T
```

Notons ici le **AS T**, en réalité obligatoire pour toute sous-requête dans un FROM.

# Sous-requêtes : exercices...

Etudiants (NumINE, Nom, Prenom, Groupe)

Matieres (NumMat, Intitule)

Epreuves (NumEpr, NumMat, date)

Notes(NumINE, NumEpr, Note)

Requêtes à écrire :

1. Afficher le nombre maximum d'étudiants dans un même groupe
2. Afficher, pour chaque étudiant de votre groupe, son nom, son prénom, sa note à l'épreuve n°8 et la moyenne du groupe à cette même épreuve
3. Afficher le classement (par ordre décroissant des moyennes générales de la promotion) des différentes matières, sachant que toutes les épreuves d'une matière ont même coefficient. Certains étudiants ont été excusés pour certaines épreuves (sinon, nous pourrions faire sans sous-requêtes : voyez-vous pourquoi ?)

## Exemple 4 : sous-requêtes de type table

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville mais, cette fois, *en affichant la ou les villes correspondantes...*

## Exemple 4 : sous-requêtes de type table et valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville mais, cette fois, *en affichant la ou les villes correspondantes...*

La même "technique" que précédemment devrait marcher mais, cette fois, la sous-requête doit également nous donner le nom des villes :

```
SELECT NomVille, COUNT(*) AS NbEtds  
FROM Etudiants  
INNER JOIN Villes ON Etudiants.NumVille = Villes.NumVille  
GROUP BY NumVille, NomVille
```



## Exemple 4 : sous-requêtes de type table et valeur

Etudiants (NumINE, Nom, Prenom, Moyenne, NumVille, Groupe)

Villes (NumVille, NomVille, CodePostal)

On veut afficher le nombre maximum d'étudiants habitant dans une même ville mais, cette fois, *en affichant la ou les villes correspondantes...*

La même “technique” que précédemment devrait marcher mais, cette fois, la sous-requête doit également nous donner le nom des villes :

```
SELECT NomVille, COUNT(*) AS NbEtds  
FROM Etudiants  
          INNER JOIN Villes ON Etudiants.NumVille = Villes.NumVille  
GROUP BY NumVille, NomVille
```

*Voyez-vous pourquoi il est nécessaire de garder **NumVille** dans le **GROUP BY** ?*

## Exemple 4 : sous-requêtes de type table et valeur

Il “suffit” maintenant d’afficher toutes les villes pour lesquelles le nombre d’étudiants est maximal... en réutilisant la sous-requête de l’exemple 3 :

```
SELECT T.NomVille, T.NbEtds  
FROM (SELECT NomVille, COUNT(*) AS NbEtds  
      FROM Etudiants  
      INNER JOIN Villes ON Etudiants.NumVille = Villes.NumVille  
      GROUP BY NumVille, NomVille) AS  
WHERE T.NbEtds =  
      (SELECT MAX(T.NbEtds)  
      FROM (SELECT COUNT(*) AS NbEtds FROM Etudiants  
          GROUP BY NumVille) AS T)
```